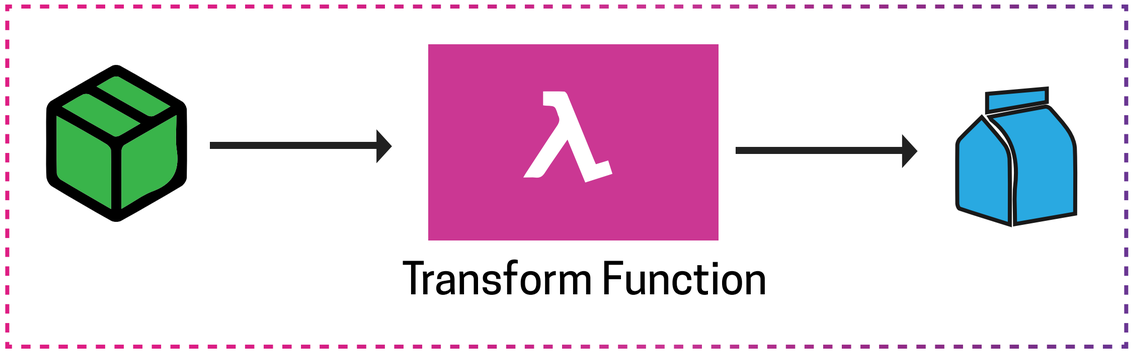
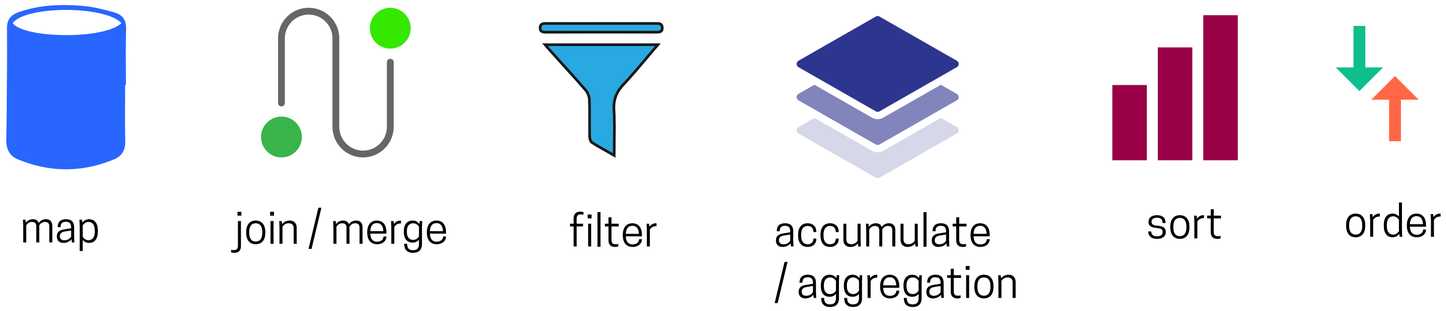
Transformation

Piping is the process of transforming data from one **state** to another **state**





Input / Output State

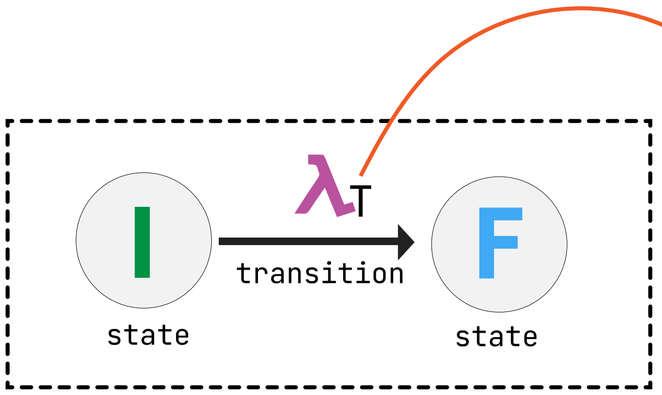
Based on the input state we can say the output state either is **deterministic** or **non-deterministic**

sum (x, y) ⟶ **x + y**

upper (str) ⟶ **STR**

greet (name) ⟶ **Hello + name**

Deterministic



Pure Function

* If the input state remains the same then the output will be the same
* Input state (params) can solely determine the output

By using this knowledge (above points)

- If the input state does not change,

- Input params can determine the output

Angular can avoid calling the **transform** function

The **transform/transition** function in the case of **deterministic** is a **pure** function derived from **FP**

A **pure** function has the property that it does not perform any **side-effects**, also hold **immutability**.

Non-Deterministic

If the output state cannot be determined from the input state only or has an **internal/external** state involved

sum (x, y) ⟶ x + y + u (unknown factor / state)

reduce (list) ⟶ 8

reduce (list) ⟶ 10

list	0xff8

0xff8

4
1
3

0xff8

4
1
3
2

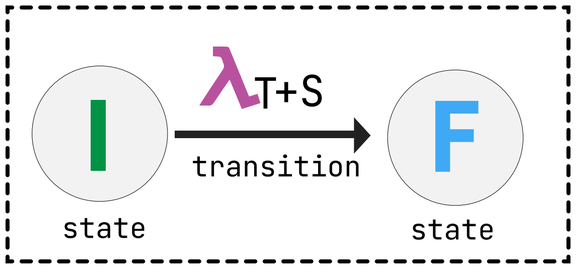
pkg (user) ⟶ 4

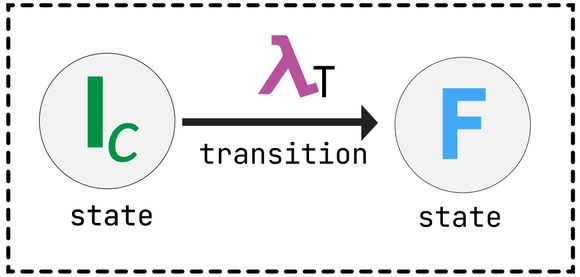
pkg (user) ⟶ 6

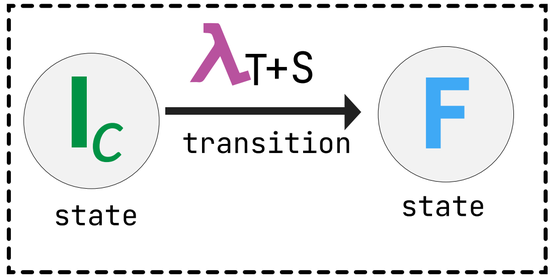
user	0xf28

0xf28 { exp: 2, ctc: 2 }

0xf28 { exp: 3, ctc: 2 }







- If Input params cannot determine the output, has internal state

-If the input state does not change, but if it is composite

Angular has to call the **transform** function (even if it is redundant)

The **transform/transition** function might be **pure** function
but if **state** is involved then it is not.

The **transform** function can have **side-effects**,
also might not hold **immutability**.

Pipe Types

Pure Pipe Deterministic

Based on the input state the output state can be determine

* Input params can solely determine the output

* If the input state remains the same then the output will be the same

Angular calls the pipe **transform** function only if **input parameters** are changed

Does not run on every change detection cycle. If input is not changed the output is same, because of deterministic nature

Pure Function

Does not perform or have any **side-effects**, also hold **immutability**.

Impure Pipe Non-Deterministic

If the output state cannot be determined from the input state only or has an **internal/external** state involved

* Input params cannot determine the output, might have internal state

* If the input state does not change, but if it is composite / reference then output can change

Angular has to call the **transform** function, even if **input parameters** are not changed (redundant calls)

Runs on every change detection cycle. There is no other way to determine the output until execute, because of non-deterministic nature

Pure/Impure Function

Can perform or have **side-effects**, might not hold **immutability**.

For data structures such as Array and Objects, if composite values changes are important then use **Impure** pipe as Angular will call the **transform** fn on every change detection cycle

Computer Baba

Ajit

Youtube Channel https://www.youtube.com/c/ComputerBabaOfficial

Twitter https://twitter.com/akacomputerbaba

Discord Server https://discord.gg/9V4VTDM